Uke 2, time 2

Fredag 29. august

Introduksjon til Python

Problemløsning

Mer kodeflyt og logiske uttrykk

Kreativitet og programmering

Egne rimelige antagelser

Mer kodeflyt og logiske uttrykk

```
dyr = "lama"
antall = 5

if 1 < antall <= 8:
    if dyr == "hest":
        print(f"{antall} hester!")
    elif dyr == "sau":
        print(f"{antall} sauer!")
    elif dyr == "lama":
        print(f"{antall} sauer!")
    else:
        print(f"Har ikke {dyr}")
else:
    print("For få eller for mange dyr")
```

```
dyr = input("Hest, sau eller lama: ").casefold()
antall = int(input("Hvor mange vil du ha? "))

if dyr == "hest" or dyr == "sau" or dyr == "lama":
    if 1 < antall <= 8:
        print(f"{antall} {dyr}er!")
    elif antall == 1:
        print(f"Én {dyr}")
    else:
        print("For få eller for mange dyr")
else:
    print(f"Har ikke {dyr}")
```

Kreativitet

Programmering er å fortelle datamaskinen hva den skal gjøre

Men det innebærer også at vi må vite hva vi vil at den skal gjøre

Situasjonen kan også være at noen ber oss om å lage et bestemt program, f.eks kan det være styringen til en brusautomat, notat-app til mobilen, eller strømmetjeneste

Men det er kanskje alt vi må forholde oss til, og da må vi gjøre antagelser

Antagelser

Når vi skal lage noe reelt, så blir vi nesten garantert ikke fortalt nøyaktig hva som skal skje i programmet. Vi får korte beskrivelser, med kun noen få konkrete krav.

Resten er det vår jobb å fylle ut

I IN1000, så vil oppgavene blir mindre og mindre spesifikke utover semesteret, så da må vi kunne bruke det vi har lært og kreativitet til å sette det sammen til noe nytt

Og hvis du er usikker, så gjør egne antagelser

Oppgaver

Oppgave 1

Programmet nedenfor mangler funksjonalitet for å legge til den største summen på slutten av setningen. Utvid programmet slik at setningen, inkludert den største summen, skrives ut. Tenk at tallene kan endre seg. Hvilke hensyn tar dere til at summene da kan være like? Diskuter og lag en plan for hva dere trenger i programmet før dere programmerer. Se kjøringen av programmet deres i Python Tutor. Det er viktig at dere forstår i detalj hva som skjer i hver programlinje samt i hvilken rekkefølge linjene kjøres.

```
tall1 = 15.7
tall2 = 9.4
sum1 = tall1 + tall2
tall3 = 37
tall4 = 11
sum2 = tall3 + tall4
setning = "Den største summen er: "
```

Oppgaver

Oppgave 2

Dere skal lage et program som simulerer en bankkonto. Brukeren skal kunne sette inn og ta ut penger fra en konto. I tillegg skal vedkommende kunne få vite saldo. Bruk én eller flere prosedyrer i programmet. Husk beskrivende utskriftssetninger underveis. Diskuter og lag en plan for hva dere trenger i programmet før dere programmerer.

Oppgave 3

Nå er det deres tur til å lage en brusautomat i Python. To kunder handler én gang hver, og under hver sin handel kjøper de én eller flere brus. Hvis brusautomaten blir tom mellom hver av handlene, eller den blir tom etter at begge kundene har handlet, skal den fylles igjen. Husk beskrivende utskriftssetninger underveis. Diskuter og lag en plan for hva dere trenger i programmet før dere programmerer

Oppgave 4

Ta utgangspunkt i oppgave 2 eller 3. Dere skal nå bevisst legge inn feil i programmet dere tar utgangspunkt i. Feilene som legges inn skal være syntaksfeil, logiske feil og kjøretidsfeil. Ta for dere én feiltype av gangen, og diskuter hvorfor valgene dere har tatt fører til de ulike feilene. Hva skiller feiltypene deres fra hverandre?